International Journal of Research and Applications
Oct - Dec © 2014 Transactions; 1(4): 137-143

ISSN (online): 2349-0020
http://ijraonline.com

ELECTRONICS ORIGINAL ARTICLE

Design and implementation of online BIST for different word sizes of memories

Ch. Priyanka [1] and S. Swathi [2]
[1] M.Tech(Pursuing) and [2] Assistant Professor
[1 2] Department of ECE, Vaagdevi College of Engineering (Autonomous),
Affiliated to Jawarlal Nehru Technological University, Bollikuntta, Warangal - 506 005.
[1] priya.rdy227@gmail.com and [2] sreeramswathi210@gmail.com

ABSTRACT

Transparent BIST schemes for RAM modules assure the preservation of the memory contents during periodic testing Symmetric Transparent Built-in Self Test (BIST) schemes skip the signature prediction phase required in traditional transparent BIST. Achieving considerable reduction in test time. Previous works or symmetric transparent BIST schemes require that a separate BIST module is utilized for each RAM under test. This approach, giver the large number of memories available in current chips, increase the hardware overhead of the BIST circuitry.

In this work we propose a Symmetric transparent BIST scheme that can be utilized to test Rams. For 5 different word widths hence, more than one RAMs can be tested in a roving manner. The hardware overhead of the proposed scheme is considerably smaller compared to the utilization of previously proposed symmetric transparent schemes.

Keywords: Built-In Self-Test (BIST), Built-In Address-Analysis (BIAA) , SRAM, BISR

1. INTRODUCTION

Memory Built-In Self-Test has become a standard industrial practice [I] - [3], since memory cores are making up a major part of the die area; it is forecasted that by 2014 they will take up 94% of the die area [4]. in addition, they are designed with minimal design rule tolerances, making them more susceptible to defects. Testing of RAM modules is performed both right after manufacturing and periodically in the field. During manufacturing testing, various kinds of tests are applied in order to ensure that the RAM operates normally. A March test comprises a series of March elements that perform a predetermined sequence of operations (read and/or write) in every word. Traditional march algorithms [5] - [7], start with an initial write-all-zero phase, where all the RAM cells are set to '0' in order to ensure that the final signature in the output compactor is known.

Periodic testing is discerned into start-up testing and testing during normal operation. Start-up testing is performed during the start-up

of the system and resembles manufacturing testing. Testing during normal operation, where the RAM normal operation is stalled (i.e. set out of normal operation), tested and then given back to operation, are applied to circuits where it is difficult and/or impractical to shut down the system since the contents of the RAM cannot be lost (e.g. space applications, wireless sensor network nodes, etc). In this kind of testing traditional march tests cannot be applied.

To deal with soft errors during system operation, adding standard online checking capabilities based on error detecting codes has been proposed [22]. Depending on the specific code, the detection of certain types of errors can be guaranteed. But, since error detection is only possible during read operations, the time between the occurrence of an error and its detection, referred to as error detection latency, may be very high. For some applications with high reliability requirements, e.g., in telecommunication switching, it is not acceptable to detect erroneous data only at the moment when the data are explicitly needed [23]. In contrast, errors should be detected as early as possible to allow for recovery before the data are requested by the system. Furthermore, error detecting codes have to increase the number of check bits to reduce the probability of masking multiple errors.

Transparent Built-in Self Test (BIST) was proposed by Nicolaidis [8]-[9] in order to confront these problems, in transparent BIST, the initial write-all-zero phase is skipped and a signature prediction phase is issued, during which a signature is captured and stored. In the sequel, a sequence of carefully selected read and write operations are performed. That leave the RAM contents equal to the initial ones; the final signature is compared to the one captured during the signature prediction phase and a decision is made as to whether a fault has occurred in the RAM or not. Transparent BIST schemes have been also proposed in [10]- [12].

One of the issues arising when transparent BIST is employed is that of the test data generator and response compactor. For traditional march algorithms the design of these two modules is trivial since known background patterns are applied (e.g. the all-O and all-i pattern), therefore a single signal is applied to the inputs of the data bus and two gates (one AND and one OR) are enough to verify the correct operation of the memory. In transparent BIST the need to capture the contents of the data contained in the memory at the beginning of the transparent test imposes the need to employ Multiple Input Shift Registers (MISR) structures, increasing the hardware overhead of the specific modules.

The idea of transparent 131ST was further evolved by Yarmolik et al [13], [14] who proposed symmetric transparent 13 1ST. In symmetric transparent 131ST, the signature prediction phase is skipped and the March series is modified in such way that the final signature is equal to the all-zero state, irrespective of the RAM initial contents. For response compaction of bit organized RAM's, in [13] a Single-Input Shift Register (SISR) was utilized whose characteristic polynomial toggles between a primitive polynomial and its reciprocal one during the different march elements of the march series. For the case of word organized RAM's it was proved that a Multiple-Input Shift Register (MISR) whose characteristic polynomial is altered in a similar fashion can serve as response compactor [14]. The scheme requires the modification of existing registers (or SISRs / MISRs) in order to serve as response evaluators and requires appropriate control logic in order to toggle between the two different polynomials during the application of the march series.

The idea of utilizing modules that typically exist in the circuit, e.g. accumulators [15j or ALU's [16], for BIST test pattern generation and/or response verification possesses advantages, such as lower hardware overhead and elimination of the need for multiplexers in the circuit path:

furthermore, the modules are exercised, therefore faults existing in them can be discovered [17]. This idea is also behind the well-known concept of software- or processor-based BIST, where instructions of a processor are applied, and using existing modules, to test the various modules of the chip [18]-[19].

In [20], [21] a Symmetric Transparent RAM 131ST scheme was proposed, where the compaction and data generation module was implemented utilizing an ALU. In circuits that contain AL Us, the output of the RAM is either directly driven to the inputs of the ALU or can be driven using processor instructions. It was shown that the scheme [211 imposes lower hardware overhead and less complexity in the control circuitry than previously proposed schemes.

On the other hand, as memory cores represent a significant portion of a multi-core chip's area (for example, Sun Microsystems' third-generation Ultra SPARC multithreaded microprocessor has 940 memories, with a total of 27 million bits) although a BIST circuit's area cost is usually low, it increases as the number of small memories in a chip grows. Therefore, reducing the required BIST circuits becomes an issue for chips containing many memory cores. In order to test memories with the same word width in a transparent way, one can use the same transparent BIST module, that have been proposed e.g. in[13]. [14]. [21] in a roving manner. However, the proposed schemes cannot be utilized to test memories having different word widths. Hence, this would require separate 131STmodulesand therefore, increased hardware overhead.

In this work we present a Symmetric Transparent Online BIST scheme for Arrays of Word-Organized RAMs (STArWaRs). The proposed scheme utilizes an ALU in order to generate the test patterns and compress the responses of the memory module; the word width of the memory can be smaller than the number of stages of the ALU. The proposed scheme can be utilized to test transparently an array of memories in a roving manner, provided that the largest width of the memory does not exceed the number of stages of the ALU. Hence, multiple Non-identical memories can be tested in a pipeline way and the area cost is drastically reduced.

The paper is organized as follows. In Section 2 a review of the previous work on March algorithms (traditional. transparent and symmetric transparent) is given. In Section 3 the proposed Symmetric Transparent Online 131ST for Arrays of Word-Organized RAMs (STArWoRs) is introduced and exemplified. In Section 4 the proposed scheme is compared to previously proposed schemes for response compaction and test data generation in symmetric transparent 131ST. Finally, in Section *5* we conclude the paper.

2. Previous Work

A march algorithm consists of n march elements, denoted by M_i. with $0 < i < n$. Each march element comprises zero(or more) write operations, denoted by w0 / w1 meaning that o / I is written to the RAM cell, and zero (or more) read operations denoted by r0 / r1, meaning that 0 / 1 is expected to be read from the memory cell. For example, the C- algorithm (Figure 1(a)) consists of six march elements denoted by M0 to M_5 [5].

In Figure 1, denotes an increasing addressing order (which can be any arbitrary addressing order) and denotes a decreasing addressing order. Traditional march algorithms erase the memory contents prior to testing; therefore, they do not serve as good platforms for periodic BIST. Nikolaidis [8] proposed the concept of transparent 131ST where the initial w0 phase is bypassed, and a the signature prediction" phase is used instead. The signature prediction phase consists of read operations and it is utilized in order to calculate a signature that will be

compared against the compacted signature calculated during the (remaining) march test, The transparent version of the C- algorithm is shown in Figure 1(b). The notation for the transparent versions of the algorithms differs from the one used in traditional march algorithms. Instead of r_0, r_1, w_0, w_1 the notations r_a, rac, wa, wac and $(r_a)^c$ arc utilized, as clarified inTable1

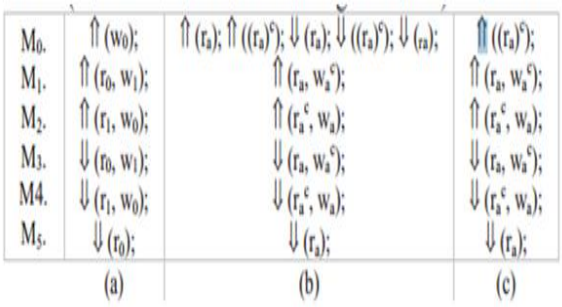

Figure 1: *C-march algorithm (a) original version, (b) transparent version, (c) symmetric transparent version*

By definition, the data driven to the compactor with the $(r_a)^c$ operation are identical to the data driven by the r_a^c. The importance of the $(r_a)^c$ operation *is* the following: during the signature prediction phase the contents of the RAM arc equal to the initial contents (since no write operation has been performed); therefore, in order to "simulate" the r_a^c.

Table1: *Notations for Symmetric transparent BIST*

Notation	Meaning
r_a	Read the contents of a word of the RAM, expecting to read the initial contents of the RAM word (i.e. before the beginning of the test)
r_a^c	Read the contents of a word of the RAM, expecting to read the complement of the initial contents of the RAM word
$(r_a)^c$	Read the contents of a word of the RAM expecting to read the initial word contents and feed the complement value to the compactor
w_a	Write to the memory word; the value that was stored in this memory word at the beginning of the test is (assumed to be) written to the word.
w_a^c	Write to the memory word; the inverse of the value that was stored in this memory word at the beginning of the test is (assumed to be) written to the word.

Operation these contents are inverted prior to entering the compactor.
It has been shown [8]. [9] That in transparent [31ST the content of the memory at the end of the test is identical to that before the test. Also, since the read elements of the 'signature prediction' phase (M_0) are identical to the read elements of the testing phase (M_1-M_5), if we store the result of the compaction of M_0 and compare it to the result of the compaction of M_1-M_5, then we can detect faults that occur due to the write operations of the march algorithm. However, traditional transparent B1ST has the disadvantage that the signature prediction phase adds up to the total testing time with a percentage of (more than) 30%.

In order to confront this problem. Yarmolik el al introduced the concept of symmetric transparent 131ST [13], [14]. They first defined the concept of symmetric data stream as follows. Let $d = (d_0, d_1,....,d_{n-1})$ $D \in \{0,1\}^n$ be a data stream: then $d^* = (d_{n-1}, d_{n-2} ..., d_1, d_0)$ denote the data stream with components in reverse order and $d^c = (d_0^c,....d_{n-1}^c)$ denote the data stream with complemented components. For example, if d=(1011), d^* = (1101) and d^c = (0100). A data string $D \in \{0, 1\}^{2n}$ is called *symmetric,* if there exists a data string $d \in \{0, 1\}^n$ with D = (d, d*) or D = (d, d^c). For example, D_1 = (1010 0101) and D_2 = (1010 1010) are symmetric data strings, since (0101) = (1010) and (1010) = $(1010)^C$ Furthermore, a transparent march rest is called symmetric if it produces a symmetric test data string D. In order to derive a symmetric transparent algorithm, the march series is modified in such way that the expected output response is equal to a known value. Therefore, the signature prediction phase can be skipped and the time required for the test is reduced.

In order to achieve this, the authors of [14] noticed that most of the march algorithms used for transparent BIST produces test data with a high degree of symmetry. For example, the read elements of the transparent C- march algorithms (Figure 1(b)), ignoring the signature prediction'

phase and the write elements, are r_a, r_a^c, r_a, In order to derive a symmetric sequence they add an additional read element, resulting in the following sequence of read elements: r_a, r_a^c, r_a. For example, for a bit-organized memory with 5 words whose initial contents are (11010), the result of the latter sequence is (00101 11010 00101 | 01011, 10100, 01011) which symmetric.

The authors of [13], [14] exploited the above-mentioned symmetry to test word-organized memories as follows. They utilized Multiple-Input Shift Registers (MISR's) and by toggling between a primitive polynomial and its reciprocal one during the up arrow ' r ' and down arrow 'r' operations, the final signature is equal to the all-zero state. In Figure 1(c) the symmetric transparent version of the C-algorithm.

The accumulator-based symmetric transparent BIST solution proposed in [20], [21] utilizes an accumulator with an one's complement adder and sterns from the observations that (a) if the march algorithm is symmetric, then the number of r_a elements equals the number of (r_a^c)elements plus the number of $(r_a)^c$ elements without taking into account the addressing order of the march element and (b) the accumulator-based compaction of the responses holds the order-independent property.

3. Proposed Scheme

In order to present the proposed scheme that can be utilized for more than one memory with varying number of word widths, in the sequel we shall present how we can perform transparent BIST when the number of stages of the ALU is larger than the memory word width.

3.1 Symmetric transparent HIST for memories with smaller word width than the number of the ALU stages

For the description of the proposed scheme, we will denote with n the number of stages of the ALU that can perform one's complement addition and with k the number of bits of the RAM word (hence k<n). The purpose of the proposed scheme is to assure that the contents of the register will be equal to a specific value (i.e. '11 ... 1') at the end of the test. in order to assure this, the (n-k) high-order inputs of the ALU are appropriately driven by the all-1 or all-0 value.

It should be noted that, if the march algorithm is symmetric, then the inputs driven to the response verifier and data It should be noted that, if the march algorithm is symmetric, then the inputs driven to the response verifier and data generator during consecutive march elements, are complementary. In order to expand this for the case where the width of the memory is smaller than the number of the ALU stages, we add a series of (n-k) 1's, i.e. a $\{1^{n-k}\}$ pattern at the high-order inputs of exact the half elements added to the ALU.This stems from the fact that if $a \le 2^k$, since

$$a + a^c = 2^k\text{-}1,$$

then we also have that

$$(\sum_{i=k}^{n-1} 2^i) + a + a^c = 2^n\text{-}1$$

For example, if k=3 and n=5, let us take the case where a = $(101)_2 = (5)_{10}$ and $a^c = (010)_2 = (2)_{10}$. Then we have that

$$(\sum_{i=k}^{n-1} 2^i) + a + a^c = 2^3 + 2^4 + 5 + 2 = 8 + 16 + 5 + 2 = 31 =$$

$$2^5\text{-}1 = 2^n\text{-}1$$

Using the above observation, we can extend the scheme proposed in [21] in order to handle the case where the ALU has more stages than the RAM word width. More precisely, we can stuff the high-order bits with a signal that has the value '1' during half the cycles and '0' during the other half. The idea is exemplified in the Figure 2.

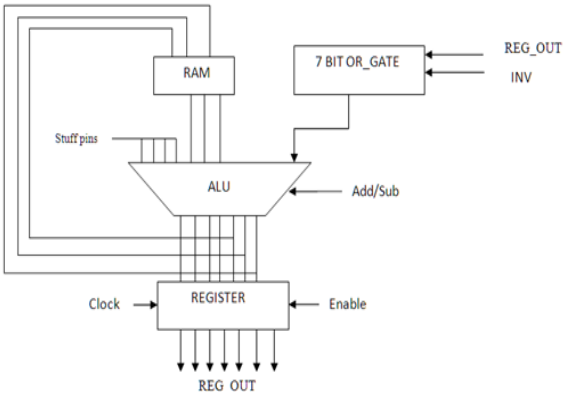

Figure 2: *The proposed scheme for transparent testing of a RAM with 3-bit words using a 7-stage ALU*

in Figure 2 we present the situation where a memory with 3- bit words is transparently tested with a 5-stage ALU. The 7- 3=4 high-order inputs of the ALU are driven by the signal Stuff. We assume that the initial contents of the RAM words arc {010, 111, 011, 100}.. The number in the parentheses denotes the address of the accessed RAM word. Depending on the operation the value of the inv signal, the contents that are written to the address (for the write operations), the value of the add/sub signal, the input to the all comprising the additional inputs (Stuff signal) and the output of the RAM (for the read operations) and the contents of the register in every cycle. The value of the signal Stuff is "1111" exactly half the times of the test. We can see that the final value of the register for the case of the fault free RAM is the all-1 value.

3.2 Transparent on line BIST for an array of RAM modules

The discussion of the previous subsection can be exploited in order to transparently test more than one memory modules, having different word widths in a roving manner. We exemplif the idea in Figure 3, for the case where three RAM modules, having words with 3, 4, and 5 bits each, respectively, are to be transparently tested on line in a roving manner using a 5-stage ALU. The RAM to be tested is enabled through the csl, cs2 or cs3 chip select signal respectively. When RAM 1 is tested, the inputs of the ALU are driven by the outputs of the RAM 1 when RAM2 is tested, the high-order input of the ALU is driven by the *S1z472* signal when RAM3 is tested, the two high-order inputs are driven by the signals *Stufli* and *Srufl2*, in a fashion similar to the one presented in Figure 2.

4. Conclusion

In this work we have presented a scheme for the testing of RAM modules using the symmetric transparent principle. The proposed scheme tests a RAM utilizing an ALU module whose number of stages can be larger than the RAM word width and naturally evolves into a scheme that can be used to test an array of RAM modules where the largest RAM word width does not exceed the number of stages of the ALU.

References

[1] R. Ailken, et. al, "A Modular Wrapper Enabling High Speed BIST and Repair for Small Wide Memories", Proc. of mt. Test Conference, pp. *997-1005*, 2004.

[2] X. Du, N. Mukherjee, W.T Cheng and S.M Reddy, "Full- speed field-programmable memory BIST architecture", Proc. of mt. Test Conference, pp. 1173-1182, 2005.

[3] X. Du, N. Mukheijee, W-T Cheng, S. M. Reddy, "A Field-Programmable Memory BIST Architecture Supporting Algorithms and Multiple Nested Loops", Proc. of the Asian Test Symposium, paper 45.3, 2006.

[4] ITRS200 1, "International Technology Roadmap for Semiconductors 2001". http://public.itrs.net/Files/20001ITRS/home.html.

[5] A. J. van de Goor, "Using March Tests to Test SRAMs", IEEE Design and Test of Computers, 1993.

[6] **A.J.** van de Goor and **C.A.** Verruijt, "An overview **of** deterministic functional RAM chip Testing", ACM computing surveys, vol. 22, no.1, Mar 1990, pp. 5-33.

[7] I. Voyiatzis, "An ALU based BIST scheme for Word- organized RAMs", IEEE Transactions On Computers, vol. 57, no. 58, May 2008.

[8] M. Nicolaidis, "Theory of transparent BIST for RAMs". IEEE Transactions on Computers, vol. 45, no. 10, October 1996, pp. 1141-1156.

[9] M. Nicolaidis, "An Efficient Built-In Self Test for functional test of Embedded RAMs", 15th Symposium on Fault Tolerant Computing, 1985.

[10] Jin-Fu Li, "Transparent-Test Methodologies for Random Access Memories Without/With ECC", Computer-Aided Design of Integrated Circuits and Systems, IEEE 1888 - 1893

[11] K. Thaller and A. Steininger. "A transparent online memory test for simultaneous Detection of functional faults and soil errors in memones." thEE Trans. Rd., vol. *52,* no. 4, pp., Dec. 2003.

[12] D.-C. Huang andW.-B Jone, "A parallel transparent BIST method for embedded memory arrays by tolerating redundant operations." IEEE Trans. Compul.-Aided sign lntegr. Circuits Syst., vol. 21, no. *5,* pp. 6l728, May2002.

[13] V. N. Yarmolik, S. Hellebrand. H.-J. Wunderlich, "Symmetric Transparent BIST for RAMs", in Germany, March 9-12, 1999.

[14] V.N. Yarmolik., I.V. Bykov, S. Hellebrand, 11.-i. Wunderlich, Transparent Word-Oriented Memory BIST based on Symmetric March Algorithms", in European Dependable Computing Conference, 1999.

[15] Voyiatzis. "Test Vector Embedding into Accumulator- generated sequences: A Linear-Time Solution", IEEE Transactions on Computers. April *2005.*

[16] A. Stroele. 'BIST Patter Generators Using Addition and Subtraction Operations", Journal of Electronic Testing: Theory and Applications, vol. 11, pp. 69-80. 1997.

[17] R. Dorsch. H.-J. Wunderlich, "Accumulator-Based Deterministic 81ST", International Test Conference, pp. 412-421, 1998.

[18] J. Zhou and H.-J. Wunderlich, "Software-Based Self Test of Proeessors under Power Constraints", Proc. Design, Auwmation and Test in Europe Conf. (DATE 06), 2006, pp 430-435.

[19] M.Psarakis, D.Gizopoulos, E.Sanchez, M.Sonza Reorda. 'Microprocessors Software Hased Set f-Testing", IEEE Design & Test of Computers Magazine, May-June 2010.

[20] I. Voyiatzis. "Accumulator-based compression in Symmetric Transparent RAM 131ST', in IEEE International Conference on Design & Technology of Integrated Systems in Nanoscale Technology. 2006.

[21] Voyiatzis, "An Accumulator - based compaction scheme with reduced aliasing for on-line BIST of RAMs", IEEE Transactions on VLSI Systems, vo. 16, no. 9, September 2008.

[22] T.R.N. Rao and E. Fujiwara, Error-Control Coding for Computer Systems.Englewood Cliffs, NJ.: Prentice Hall Inc, 1989

[23] S. Barbagallo, D. Medina. F. Corno, P. Prinetto, and M. Son,a Reorda. integrating Online and Omine Testing of a Switching Memoiy,° IEEE Design & Test of Computers. vol. ***15,*** no. 1, pp. 63-70. Jan.-Mar. 1998.